

Métodos de Desenvolvimento de Software (MDS) 2011/2012

Miguel Goulão
mgoul@fct.unl.pt
<http://ctp.di.fct.unl.pt/~mgoul/>

2

Análise de Requisitos

Modelo de casos de uso (Use Cases)

Modelo de casos de uso (*use cases*)

(Análise de Requisitos)

3

- Um **caso de uso** descreve **como** o sistema será usado
 - ▣ descreve a essência do sistema na perspectiva dos utilizadores
- Um diagrama de casos de uso **ilustra** essa essência **visualmente**
- Um modelo de casos de uso **descreve** a essência do sistema segundo a visão dos utilizadores
 - ▣ descrição de casos de uso
 - ▣ definição das interações entre o sistema e os actores
- Meio de comunicação entre o engenheiro de software e o utilizador

Modelo de use cases: sistema, actores e casos de uso

4

- Decomposição funcional do sistema em casos de uso e respectivos actores
 - ▣ Casos de uso representam os requisitos dos utilizadores
 - ▣ Actores são entidades externas que interagem com o sistema
- Construção do modelo de casos de uso
 - ▣ diagrama de casos de uso global
 - ▣ descrição detalhada de cada caso de uso

Modelo de use cases: conceitos

5

- Actor: papel desempenhado por um utilizador, *software* ou *hardware*.
- Use case (caso de uso):
 - ▣ diálogo que mostra uma sequência de transacções entre o sistema e o utilizador
 - ▣ descreve cenários de utilização do sistema
 - ▣ foca uma funcionalidade completa (visão optimista e situações de erro)
- Os casos de uso são usados no levantamento de requisitos; eles constituem a base para a análise, desenho, testes e implementação.

Modelo de use cases: um exemplo

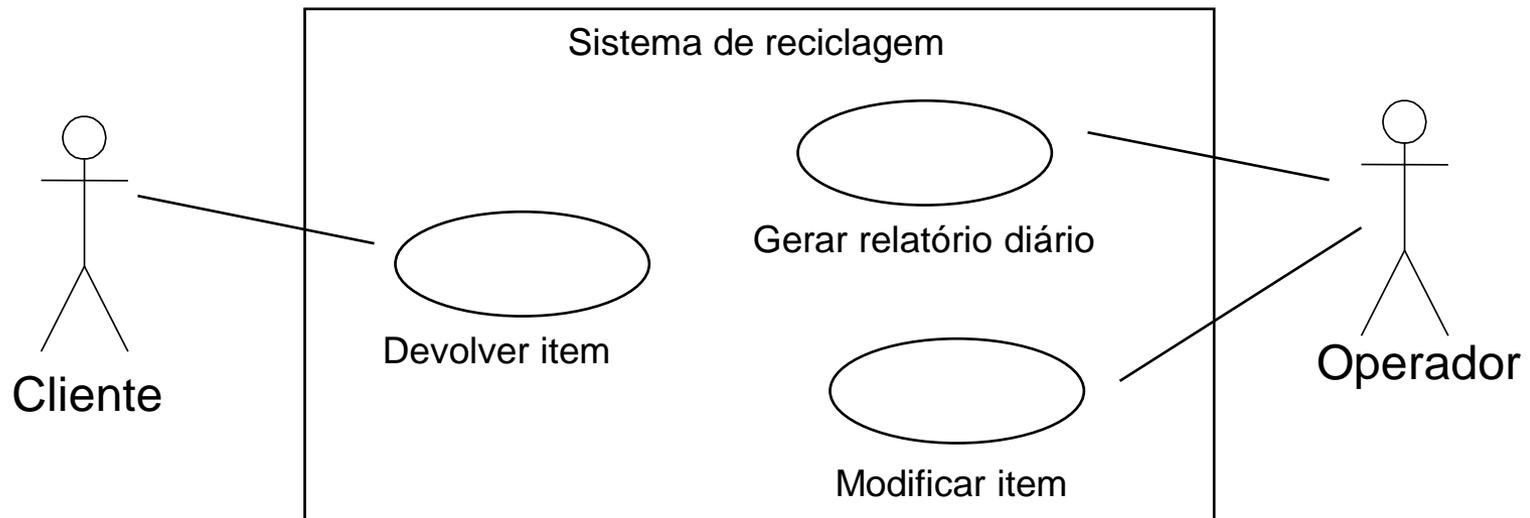
6

- Considere um sistema de reciclagem de garrafas, latas e grades. Como cada item tem dimensões e preços diferentes, o sistema tem que identificar que tipo de item acabou de receber. O sistema regista o número de itens e, se o cliente pedir um recibo, imprime o número de itens devolvidos, o seu tipo, os preços parciais e o total que será depois pago ao **cliente** → **Actores**
- O sistema também é usado por um **operador** que, ao fim do dia, pede uma listagem dos itens devolvidos nesse dia.
- O operador pode ainda mudar informação no sistema.

Quais os actores e os casos de uso?

7

- Actores: cliente e operador.
- O cliente pode:
 - ▣ Devolver itens. Isto forma o use case “Devolver Item”.
 - Este use case inclui tudo desde a devolução do item até à impressão do recibo.
- O operador pode:
 - ▣ Pedir listagem diária. Isto forma o use case “Gerar relatório diário”.
 - ▣ Modificar informação. Isto forma o use case “Modificar item”.



Descrever casos de uso

8

- Cada caso de uso TEM que ser descrito mais detalhadamente
- Eis uma descrição textual para “Devolver item” (**complete você com a descrição dos casos de erro**):
 - ▣ Devolver item é iniciado pelo cliente quando quer devolver garrafas, latas ou grades. Por cada item inserido na máquina, o sistema incrementa o número de itens devolvidos pelo cliente e ainda o número diário total de itens daquele tipo. Quando o cliente já não tem mais itens para devolver, pede um recibo carregando no botão de recibos. O sistema faz os cálculos necessários e, por cada tipo de item devolvido, calcula o preço e o número de itens devolvidos. Esta informação é imprimida, um item por linha, por uma impressora. Finalmente o sistema calcula o valor total e a impressora imprime-o. Sempre que ...**<complete com as situações de erro>**

Modelo de casos de uso

9

- Casos de uso são simples (são visualmente simples), mas identificar os que interessam e descrevê-los correctamente é difícil
- O modelo de casos de uso é o modelo mais importante que você pode criar
 - Porque explicitar, reconhecer e organizar os objectivos é fundamental para os conseguir atingir
 - “Uma viagem de milhares de Km começa com um passo” [Lao Tzu]
 - “Se você não sabe onde vai, a jornada não terá fim”

Descrever casos de uso

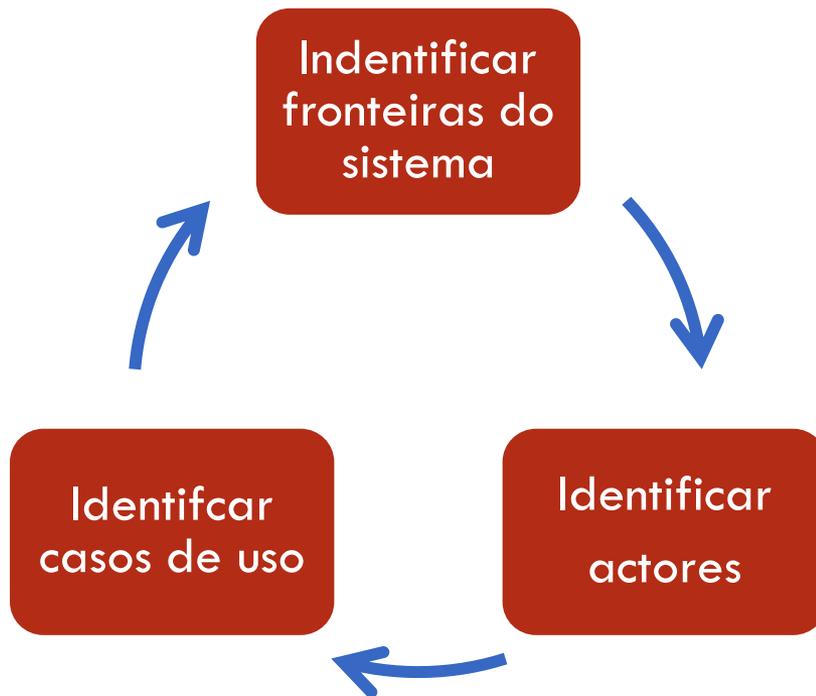
10

- Qual a diferença entre um caso de uso e um método/operação de uma classe?
- Formas de identificar casos de uso
 - ▣ Top-down
 - ▣ Bottom-up
- Qual a forma mais adequada?
 - ▣ Enunciado do problema
 - ▣ Organização mental e capacidade de abstracção de cada um
 - ▣ Consistência na granularidade

Modelação de casos de uso

11

Processo

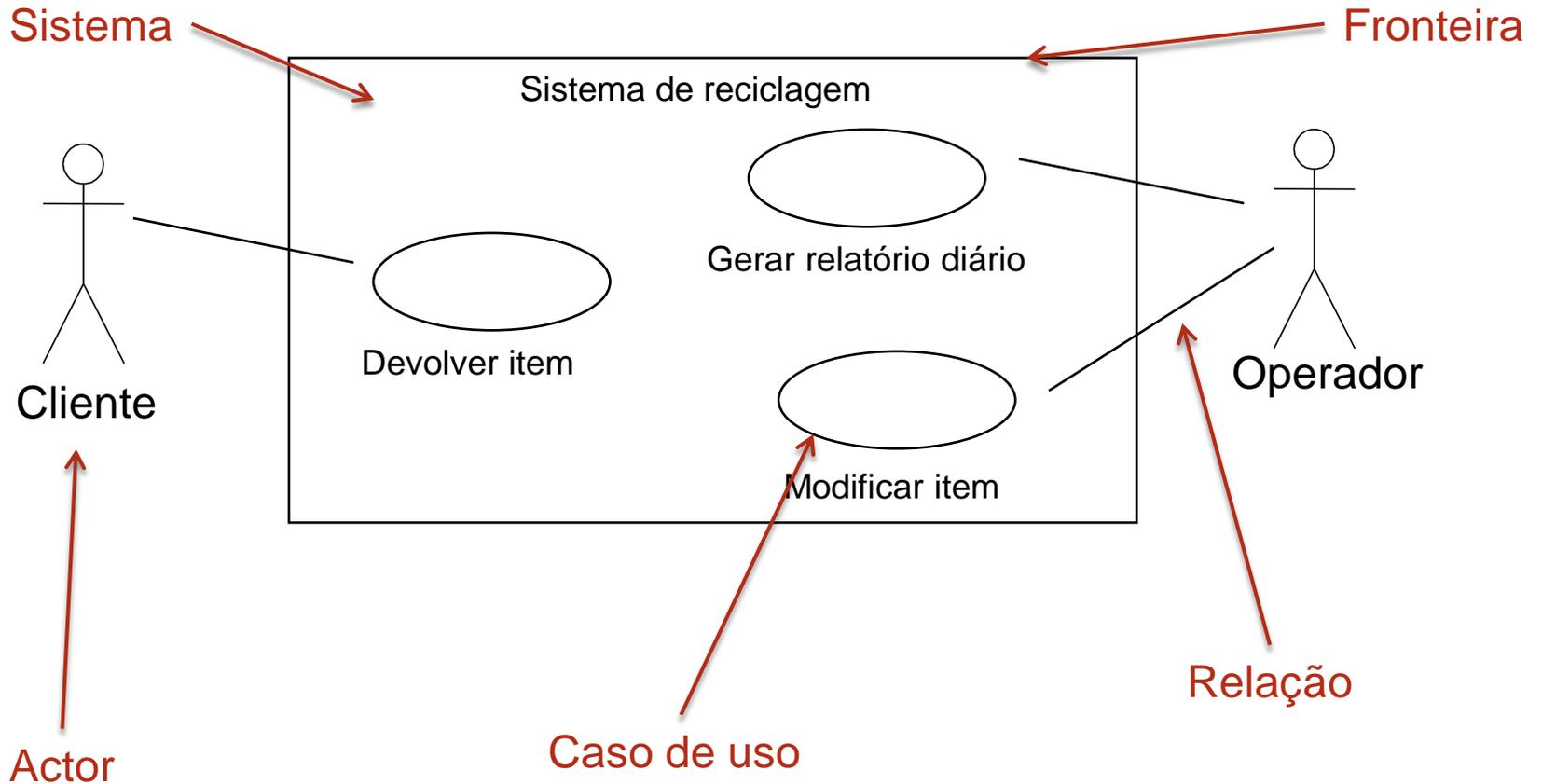


Resultados do processo

- Fronteiras do sistema
- Actores
- Casos de uso
- Relações entre actores e casos de uso

Diagrama de casos de uso

12



Sistema (definição de fronteiras)

13

- Definir as fronteiras do sistema
 - ▣ O que é que faz parte do sistema?
 - ▣ O que é que não faz parte do sistema?
- Sistema desenhado como um rectângulo
 - ▣ Actores desenhados **fora** do sistema
 - ▣ Casos de uso desenhados **dentro** do sistema

Parece trivial, mas falhar esta definição pode significar (e muitas vezes significa mesmo) o fracasso do projecto!

A fronteira tem um impacto enorme na definição dos requisitos funcionais e não funcionais do sistema.



Actores

14

- Um actor especifica um **papel** que uma entidade externa ao sistema adopta ao interagir directamente com o sistema
- Actores interagem com o sistema e podem ser
 - Pessoas
 - Outros sistemas
 - Dispositivos de hardware

Não confunda o papel que uma entidade desempenha num sistema com a entidade em si. O mesmo papel pode ser desempenhado por várias entidades.

Identificação de actores



15

- Perguntas que ajudam a identificar actores:
 - ▣ Quem (ou o quê) necessita do sistema para poder realizar suas actividades?
 - ▣ Que papel representa na interacção com o sistema?
 - ▣ Quem está interessado nos resultados do sistema?
 - ▣ Quem é responsável pela administração do sistema?
 - ▣ Com que outros sistemas o sistema deve comunicar?
 - ▣ Quem fornece informação ao sistema?
 - ▣ Acontece alguma coisa num determinado momento?

Algumas regras na modelação de actores



16

- Os actores são sempre externos ao sistema
- Os actores interagem directamente com o sistema
- Os actores representam um papel no sistema, não pessoas ou coisas específicas
- Uma pessoa, ou coisa, pode representar mais que um papel, sendo representada por mais que um actor no sistema
- O actor deve ter sempre um nome e uma descrição curta que o descrevem na perspectiva do negócio
- O tempo pode ser representado como um actor, se periodicamente houver um evento despoletado (por exemplo, a gravação periódica de documentos num processador de texto)

Actores

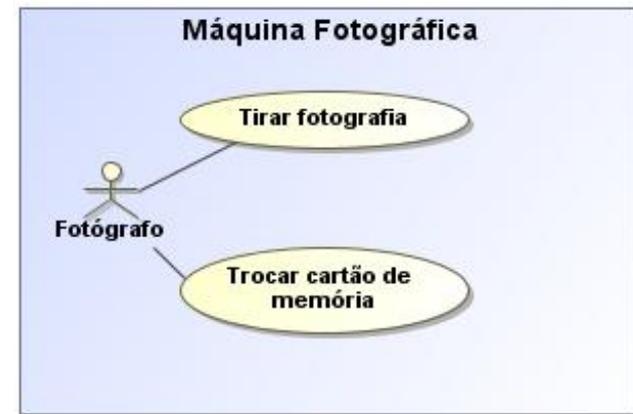
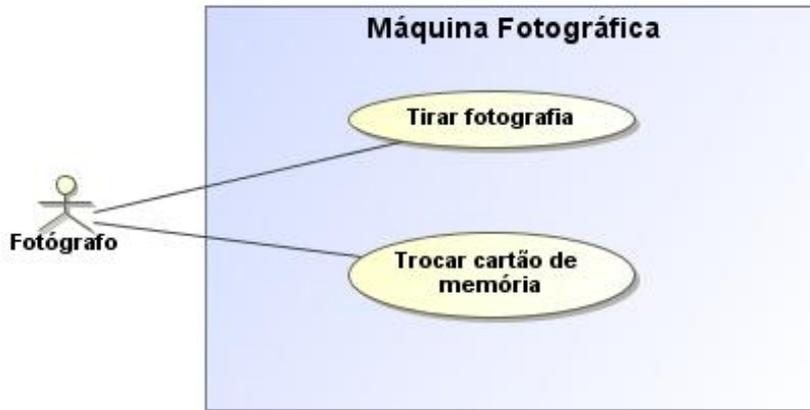


17

- Os **actores** estão sempre **fora do sistema** (os **casos de uso** estão sempre **dentro**) e portanto são **sempre** representados fora das fronteiras.

Correcto

Incorrecto

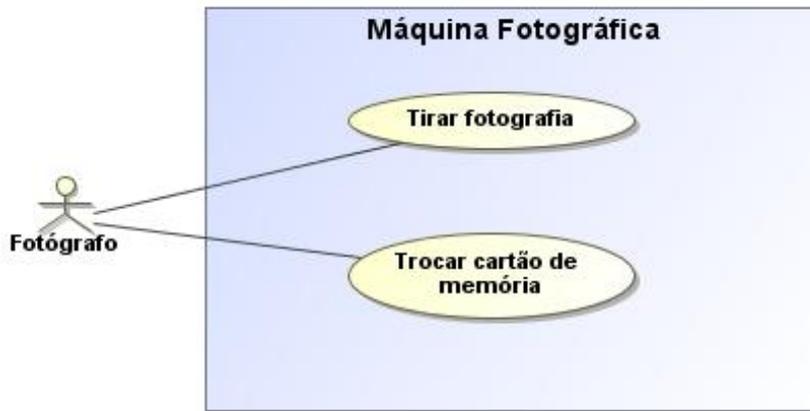


Actores

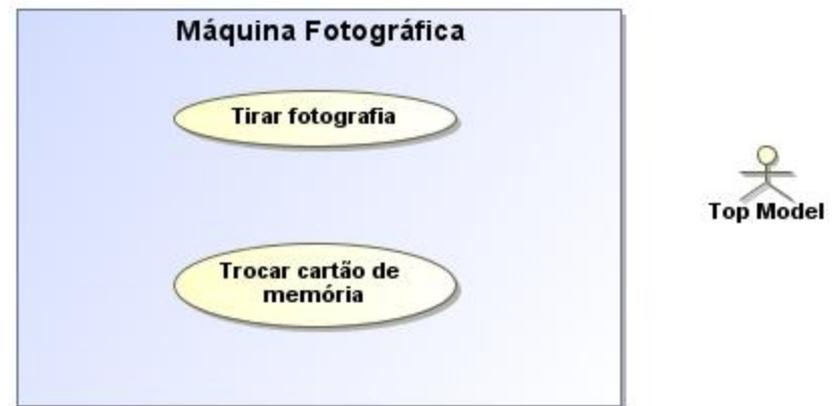
18

- Cada actor deve comunicar directamente com pelo menos um caso de uso do sistema. Caso contrário, faz mesmo sentido considerar esse actor? Não!

Correcto



Incorrecto

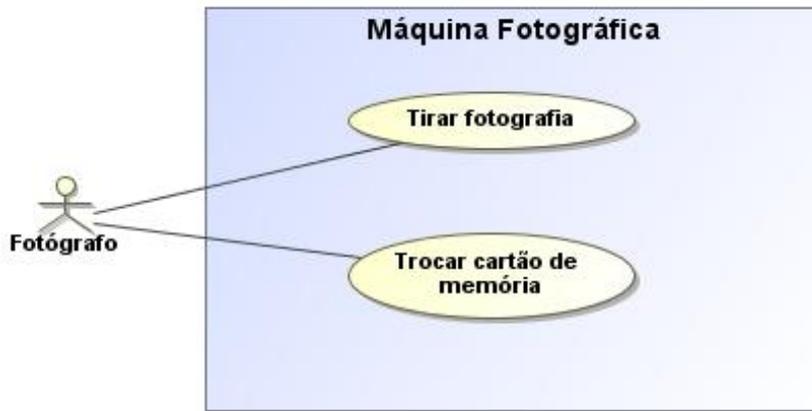


Actores

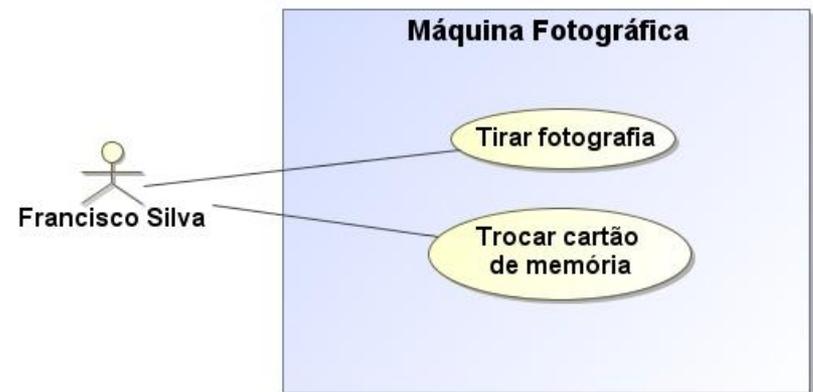
19

- Os actores representam um papel no sistema, e não pessoas ou coisas específicas

Correcto



Incorrecto



Actores

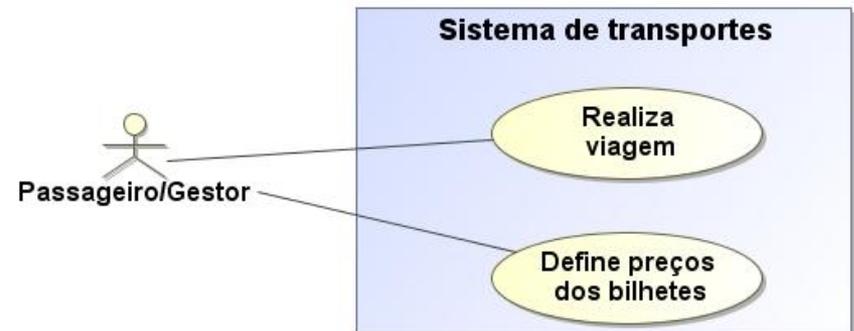
20

- Uma pessoa, ou coisa, pode representar mais que um papel, sendo representada por mais que um actor no sistema
 - ▣ Ex. A mesma pessoa pode ser passageiro e gestor de transportes públicos)

Correcto



Incorrecto



Actores

21

- O actor deve ter sempre um nome que faça sentido na perspectiva do negócio
- O actor deve ter uma descrição curta que o descreve na perspectiva do negócio

Correcto

Nome: Passageiro

Descrição: Passageiro do transporte público

Incorrecto

Nome: Pessoa

Descrição: Pessoa que se desloca entre vários pontos da cidade para tratar de certos e determinados assuntos irrelevantes para o sistema, mas que descrevemos aqui só porque sim...

Actores

22

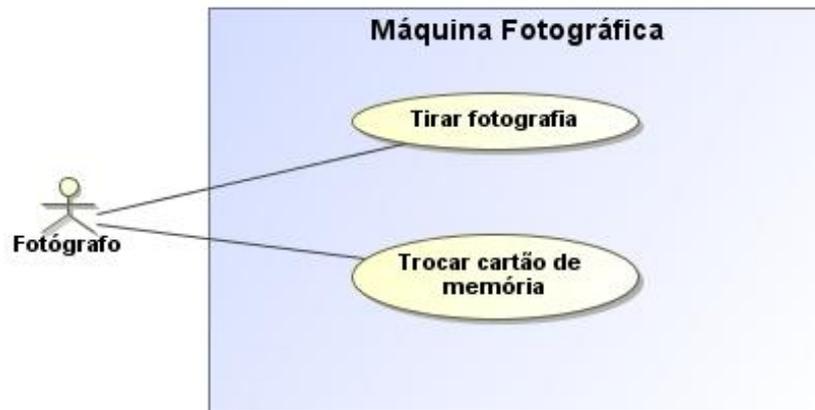
- O tempo, como actor
 - ▣ Por vezes, queremos modelar coisas que acontecem no sistema num momento bem determinado no tempo, mas que não parecem ser despoletadas por nenhum actor em particular
 - ▣ Nesses casos, podemos criar um actor especial Tempo
 - Exº: “Mensalmente, o sistema de email apaga todas as mensagens classificadas como spam”

O que colocar dentro do sistema num diagrama de casos de uso?

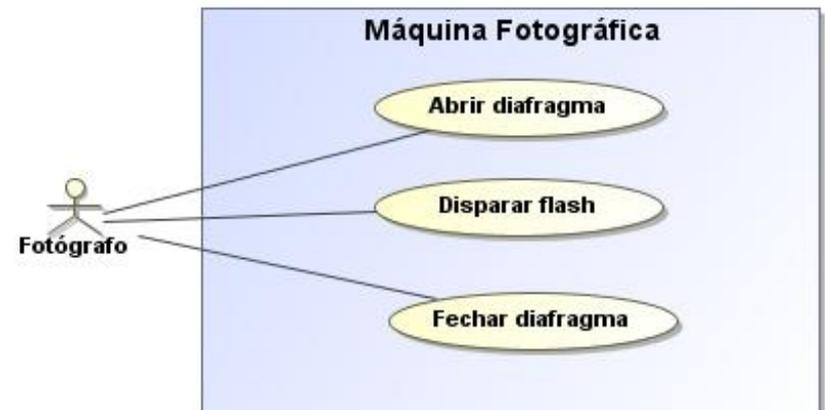
23

- Devemos colocar apenas casos de uso de alto nível. Comportamentos internos do sistema não devem ser representados aqui.

Correcto



Incorrecto



O que colocar dentro do sistema num diagrama de casos de uso?

24

- A “caixa” que representa o sistema aparece apenas nos diagramas de topo, mas é normalmente omitida em sub-diagramas.
- Comportamentos internos do sistema que são apenas usados por outros elementos do sistema não devem aparecer nos diagramas de topo.

Se um caso de uso representa um serviço de alto nível, deve fazer sentido que um actor possa usar apenas esse serviço durante uma “sessão” de utilização do sistema.

Casos de uso - resumo

25

- O conjunto de todos os casos de uso descreve todas as maneiras possíveis de usar o sistema
- Começando por identificar actores, é mais fácil depois identificar os casos de uso. Cada actor executa um ou mais casos de uso
- Um caso de uso representa-se graficamente por uma oval e descreve-se textualmente em linguagem natural



Identificar casos de uso

26

- Se possível, entrevistar os utilizadores do (futuro) sistema
- Interpretar os requisitos na perspectiva dos actores
- Ajuda a responder às seguintes perguntas:
 - ▣ Quais são as tarefas principais de um actor?
 - ▣ Será que o actor tem que ler, criar ou mudar informação do sistema?
 - ▣ Será que o actor tem que informar o sistema sobre alguma mudança externa?
 - ▣ Será que o actor recebe alguma informação do sistema?
 - ▣ Será que o actor quer ser informado sobre mudanças inesperadas?

Modelo de casos de uso

27

- Mostra as fronteiras do sistema e capta a funcionalidade que o futuro sistema deve oferecer
- Pode funcionar como um entregável entre o engenheiro de software e o cliente
 - ▣ Por isso, é importante que seja legível por não especialistas em UML
- É a base para as fases seguintes
 - ▣ É estruturado pelo modelo de análise

28

Exemplo

Sistema de marcação de consultas médicas

Fazer um diagrama de casos de uso (UC) para marcação de consultas médicas

29

- Num sistema de gestão de consultas médicas, um paciente em contacto com um responsável pelo mapa de marcações marca a sua consulta. Eventualmente ela pode ser cancelada por indisponibilidade do paciente.
- Durante a consulta o médico pode receitar algum medicamento para o paciente. Esta informação deve ser guardada para uma futura consulta.
- Finalmente o paciente deve pagar a consulta a um empregado do consultório médico, onde obterá um recibo.

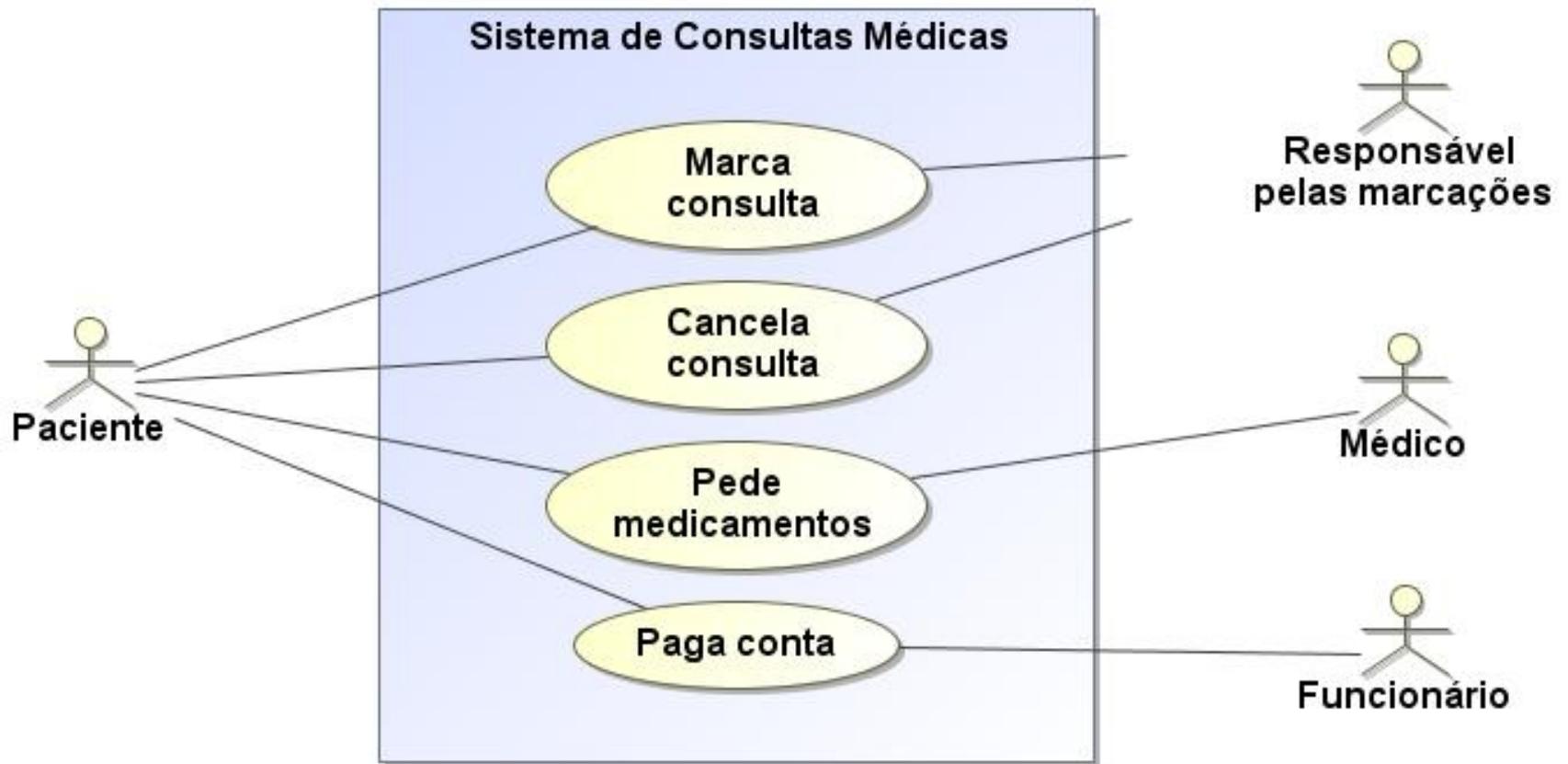
Actores do exemplo?

30

- Paciente
 - ▣ Marca consulta
 - ▣ Cancela consulta
 - ▣ Pede medicamentos
 - ▣ Paga conta
- Responsável pelo mapa de marcações
 - ▣ (Marca consulta, como actor secundário)
 - ▣ (Cancela consulta, como actor secundário)
- Médico
 - ▣ (Pede medicamentos, como actor secundário)
- Empregado do consultório médico
 - ▣ (Paga conta, como actor secundário)

Modelo de UCs para Consultas Médicas

31



O que representam os casos de uso?

33

- Um **caso de uso** é composto (definido) por um **conjunto de cenários**
- Cada cenário contém uma sequência de passos que descreve uma interacção entre o utilizador e o sistema
- Um caso de uso reúne cenários que satisfazem um determinado **objectivo** do utilizador

Como descrever use cases?

34

- Descrevendo os seus cenários: em geral um caso de uso tem um **cenário principal** (ou primário) e vários **cenários alternativos** (ou secundários)
- Usando descrições semi-estruturadas: templates
- Textualmente
- Usando técnicas formais: expressões matemáticas, linguagens algébricas, etc.
- Usando outros modelos UML: diagramas de actividade e diagramas de sequência

Cenários

35

- Um cenário é uma sequência de eventos mostrando as interacções típicas e a informação trocada entre um utilizador e o sistema.
 - ▣ Ocorre durante uma execução particular do sistema
 - ▣ Pode ser escrito em linguagem natural
 - ▣ É uma instância específica de um caso de uso
- Qualquer use case tem sempre 2 tipos de cenários:
 - ▣ **Principal** (ou primário): descreve a situação quando tudo corre bem (visão optimista – *happy day scenario*)
 - ▣ **Secundário** (ou alternativo): permite uma sequência diferente de eventos em relação ao cenário principal, tratando excepções e casos de erro

Template: uma proposta

36

Nome: nome do caso de uso

Descrição: descrição executiva

Actores: que comunicam com o caso de uso

Principais

Secundários

Pré-condições: pré-requisitos para a execução com sucesso

Cenário principal: passos atómicos do caso de uso

Cenários secundários: desvios do cenário com sucesso

Pós-condições: estado do sistema depois da execução bem sucedida

Especificação de casos de uso

37

- Nome
 - ▣ Escolha nomes curtos, mas descritivos
 - ▣ O nome deve ser claro para que um cliente, não especialista em MDS, seja capaz de ficar com uma ideia clara de que função, ou processo de negócio, está a ser implementada pelo caso de uso, só de ler o nome do caso de uso

Especificação de casos de uso

38

□ Descrição

- Deve conter apenas um parágrafo resumindo o objectivo do caso de uso.
- Deve capturar a essência do caso de uso, ou seja, o “benefício de negócio” para cada um dos actores envolvidos

Especificação de casos de uso

39

- Actores (use os mesmos nomes do diagrama de UCs
 - Principais: os iniciadores do caso de uso
 - Qualquer actor que possa iniciar o caso de uso é considerado actor principal
 - Secundários: actores que interagem com o caso de uso, embora não o iniciem
 - Qualquer actor que participe no caso de uso, mas não o possa iniciar, é considerado actor secundário
- Cada caso de uso é iniciado por apenas um actor
- Actores diferentes podem iniciar o mesmo caso de uso
- Se o caso de uso não tiver actores secundários, é bom estilo escrever “Nenhum” na respectiva secção

Especificação de casos de uso

40

- **Pré-condições:** restringem a utilização do sistema **antes** de o caso de uso se iniciar
 - ▣ O caso de uso não se pode iniciar se elas forem falsas
- **Pós-condições:** restringem o estado do sistema **depois** de o caso de uso se executar
 - ▣ Especificam o que é verdade após o final da execução do caso de uso
- Se o caso de uso não tiver pré, ou pós-condições, deve escrever “Nenhumas” na respectiva secção
 - ▣ Deixar em branco é ambíguo; quem lê o documento deve perceber que considerou o assunto e decidiu não estabelecer pré ou pós condições

Especificação de casos de uso

41

- **Cenário principal:** Descreve o caso de uso, passo a passo
 - ▣ Os passos devem ser numerados
 - ▣ O cenário começa sempre com um actor principal a iniciar o caso de uso, ao desempenhar uma qualquer função.
 - O caso de uso começa quando o <actor><função>
 - ▣ Cada passo deve ser numerado e ter o formato:
 - <número> O <actor/coisa> <qualquer acção>.

Especificação de casos de uso

42

- Bons exemplos:
 - ▣ 1. O caso de uso começa quando o cliente selecciona a operação “Fazer encomenda”
 - ▣ 2. O cliente preenche o seu nome e morada no formulário

Em ambos os casos, temos frases simples e declarativos em que fica claro que alguém ou alguma coisa realiza uma acção.

Especificação de casos de uso

43

- Mau exemplo:
 - “Os dados do cliente são preenchidos”

Este exemplo tem vários problemas:

- Regra geral, devemos evitar a voz passiva na formulação destas especificações
- Não sabemos:
 - Quem** preencheu os dados?
 - Que dados** foram preenchidos?
 - Onde** é que os dados ficaram preenchidos?
- Devemos poder responder de forma clara e específica a questões como:
 - Quem?**
 - O quê?**
 - Quando?**
 - Onde?**

44

Voltando às consultas médicas...

Especificação de casos de uso



45

Caso de Uso: Marca consulta

Descrição: O paciente marca uma consulta médica no consultório

Actor principal: Paciente

Actores secundários: Responsável pelas marcações

Pre-condições: Nenhuma

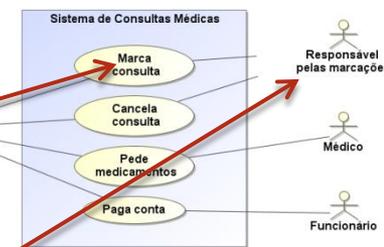
Fluxo principal:

1. O caso de uso começa quando o paciente contacta o sistema para marcar uma consulta
2. O sistema envia o pedido de consulta para o responsável pelas marcações
3. O responsável pelas marcações consulta as datas disponíveis para consulta no sistema
4. O responsável pelas marcações selecciona uma data para a consulta no sistema
5. O sistema envia a data de consulta para o paciente
6. O paciente aceita a data de consulta proposta

Pos-condições: a consulta fica marcada para a data acordada com o paciente

Fluxos alternativos: Nenhum

Especificação de casos de uso



46

Caso de Uso: Marca consulta

Descrição: O paciente marca uma consulta médica no consultório

Actor principal: Paciente

Actores secundários: Responsável pelas marcações

Pre-condições: Nenhuma

Fluxo principal:

1. O caso de uso começa quando o **paciente** contacta o **sistema** para **marcar uma consulta**
2. O sistema envia o pedido de consulta para o responsável pelas marcações
3. O responsável pelas marcações consulta as datas disponíveis para consulta no sistema
4. O responsável pelas marcações selecciona uma data para a consulta no sistema
5. O sistema envia a data de consulta para o paciente
6. O paciente aceita a data de consulta proposta

Pos-condições: a consulta fica marcada para a data acordada com o paciente

Fluxos alternativos: Nenhum

Métodos de Desenvolvimento de Software (MDS) 2011/2012

Miguel Goulão
mgoul@fct.unl.pt
<http://ctp.di.fct.unl.pt/~mgoul/>

Enriquecendo a descrição dos cenários

Exemplo motivador

Criação de ramos na descrição

Decisões

Repetições

Processamento de encomendas

Nota: Como trabalhamos para um mercado cada vez mais global, este problema e sua resolução são especificados em inglês

Descrição do problema de processamento de encomendas

50

- Problem Description
 - We are developing an order processing software for a mail order company called National Widgets, which is a reseller of products purchased from various suppliers.
 - Twice a year the company publishes a catalog of products, which is mailed to customers and other interested people.
 - Customers or their representatives purchase products by submitting a list of products with payment to National Widgets. National Widgets fills the order and ships the products to the customer's address.
 - The order processing software will track the order from the time it is received until the product is shipped.
 - National Widgets will provide quick service . They should be able to ship a customer's order by the fastest, most efficient means possible.
 - Customers may return items but will sometimes pay a fee.
- Assumptions
 - An electronic interface, such as web, would be good for some customers.
 - We expect to use multiple shipping companies and insured methods.

Actores do exemplo

51

- Customer; Customer Rep
 - ▣ a person who orders products from National Widgets
- Shipping company
 - ▣ such as DHL, FedEx
- Clerk
 - ▣ an employee of National Widgets who packages, labels, and ships orders
- Inventory system
 - ▣ software that tracks the company inventory
- Accounting system
 - ▣ software that keeps the company books

Casos de uso relacionados com a satisfação de encomendas

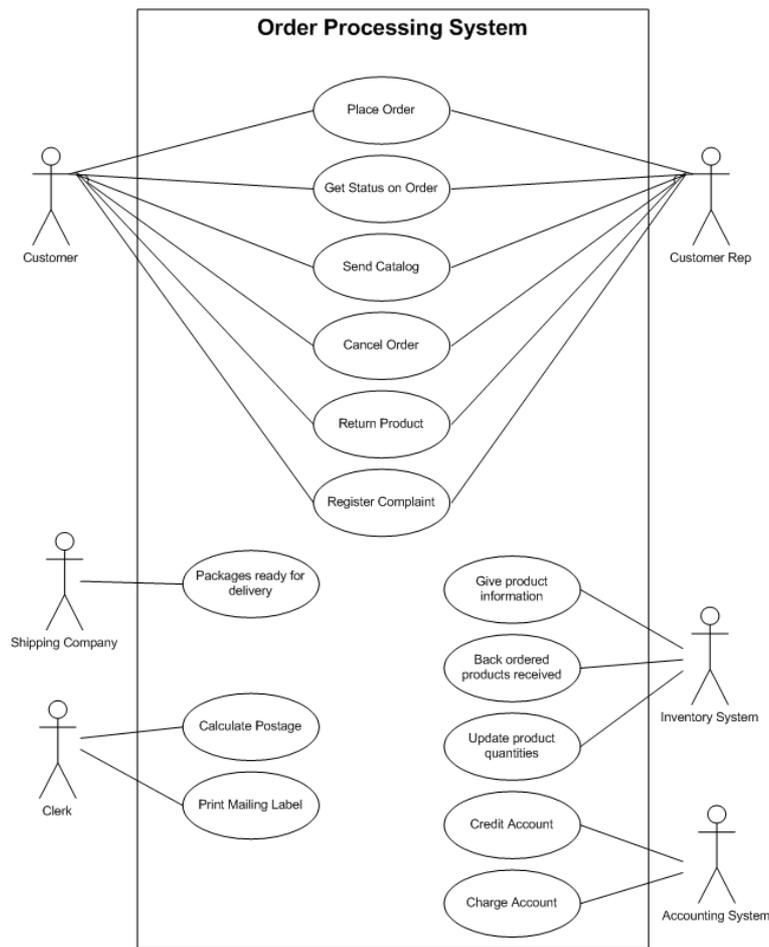
52

- From **customer** and **customer rep**: **place order, send catalog, get status on order, return product, cancel order, register complaint**
- To shipping companies: **packages ready for delivery**
- From clerk: **print mailing labels, calculate postage**
- To inventory system: **give product information, update product quantities**
- From inventory system: **back-ordered items received**
- To accounting system: **charge account, credit account**

Temos de descrever os casos de uso, primeiro textualmente, e depois, mais rigorosamente, usando cenários, diagramas de actividade, diagramas de sequência, ...

Modelo de casos de uso do sistema de encomendas

53



Comentários?

Place Order Use Case

54

Pre-condition:

A valid user has logged into the system

Flow of Events:

Basic Path:

1. The use case starts when the customer selects place order
2. The customer enters his/her name and address
3. If the customer enters only the zip code the system will supply the city and the state
4. The customer will enter product codes for desired products
5. The system will supply a product description and the price for each item
6. The system will keep running total of items ordered as they are entered

7. The customer will enter credit card payment information
8. The customer will select submit
9. The system will verify the information, save the order as pending, and forward the payment information to the accounting system
10. When payment is confirmed, the order is marked Confirmed, an order ID is returned to the customer, and the use case ends

Post Condition:

The order has been saved in the system and marked Confirmed

Place Order Secondary Scenarios

55

- payment not there
- order incomplete
- order gets lost
- shipping address is incomplete
- product code does not match actual products
- product no longer carried
- payment bad
- customer pays by check
- customer sends order by mail
- customer phones in order

Cenário secundário

56

- Cenário secundário “*Product code does not match actual products*”

1. The system shows a message saying that the product code is invalid
2. Return to step 4 of the main scenario

- “Place order” extension points: “Product code does not match actual products”, antes do passo 5

Especificar Alternativas (if-statement)

57

| |
|--|
| Caso de Uso: Saber o estado de uma encomenda |
| Descrição: O cliente procura e visualiza o estado de uma encomenda no sistema |
| Actor principal: Cliente |
| Actores secundários: Nenhum |
| Pre-condições: Nenhuma |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o cliente tenta saber o estado de uma encomenda2. O cliente preenche os dados com o identificador da encomenda, o identificador do cliente, ou o nome do cliente3. Se o cliente introduziu o identificador da encomenda<ol style="list-style-type: none">a) O sistema mostra uma ordem4. Caso contrário, se o cliente introduziu o nome do cliente, ou o seu identificador<ol style="list-style-type: none">a) O sistema retorna uma lista com todas as encomendas desse clienteb) O cliente selecciona uma das encomendas da listac) O sistema mostra a respectiva encomenda |
| Pos-condições: Nenhuma |
| Fluxos alternativos: Nenhum |

Especificar Repetições (ciclos - for)

58

Flow of Events:

Basic Path:

1. The use case starts when the customer selects place order
2. The customer enters his/her name and address
3. If the customer enters only the zip code the system will supply the city and the state
4. The customer will enter product codes for desired products
5. **For each product code entered**
 - a) **The system will supply a product description and the price for each item**
 - b) **The system will add the price of the item to the total**
6. The customer will enter credit card payment information
7. The customer will select submit
8. The system will verify the information, save the order as pending, and forward the payment information to the accounting system
9. When payment is confirmed, the order is marked Confirmed, an order ID is returned to the customer, and the use case end

Especificar repetições (ciclos - while)

59

Flow of Events:

Basic Path:

1. The use case starts when the customer selects place order
2. The customer enters his/her name and address
3. If the customer enters only the zip code the system will supply the city and the state
4. The customer will enter product codes for desired products
5. **While the customer enters product codes**
 - a) The system will supply a product description and the price for each item
 - b) The system will add the price of the item to the total

6. The customer will enter credit card payment information
7. The customer will select submit.
8. The system will verify the information, save the order as pending, and forward the payment information to the accounting system
9. When payment is confirmed, the order is marked Confirmed, an order ID is returned to the customer, and the use case end

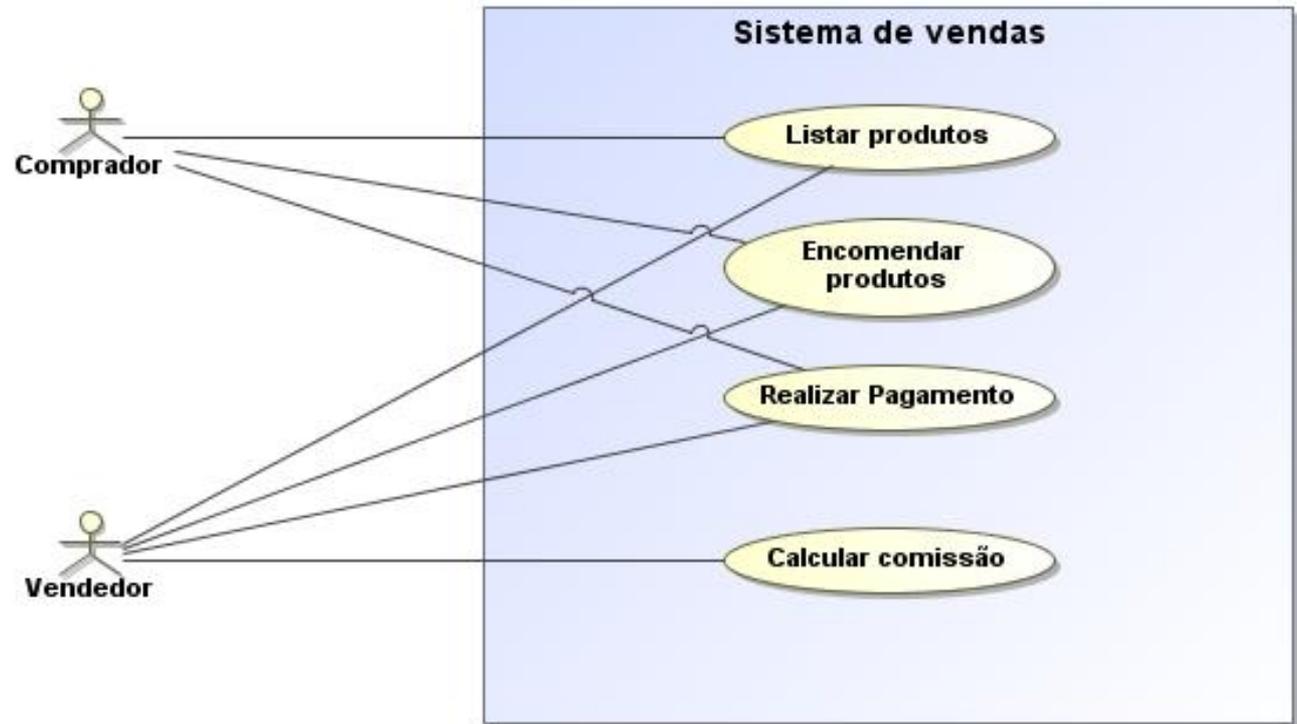
60

Generalização entre actores

Num sistema de vendas...

61

- Suponha que o comprador e o vendedor têm vários casos de uso em comum



Muito comportamento comum entre estes dois actores. Podemos simplificar o modelo...

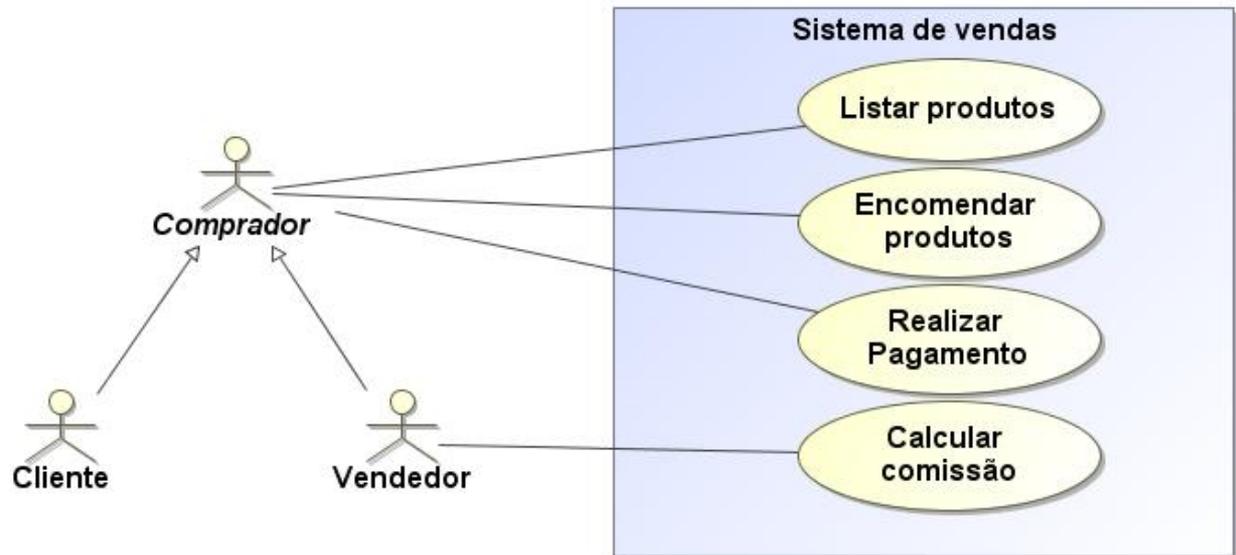
Associação “inherits”

62

- A associação herda entre atores
 - ▣ Abstracto/concreto: um actor concreto define-se a partir de um actor abstracto
- Vantagens: simplificar os diagramas

Será que poderíamos fazer vendedor herdar de cliente?

Seria um mau estilo de modelação. Em geral, é preferível manter o actor “pai” abstracto, para simplificar a semântica da herança.



Generalização entre actores: quando usar

63

- A presença de vários casos de uso partilhados por actores diferentes que no fundo representam um papel comum nesses casos de uso sugere a possibilidade de **generalizar** um actor.
 - ▣ Note que para a generalização fazer sentido, os vários actores têm de comunicar com o caso de uso da mesma forma.
- Tipicamente, é preferível manter o actor generalizado abstracto – estamos a generalizar um papel comum. A semântica da generalização fica mais simples.
 - ▣ Os actores descendentes herdam os papéis e as relações com os casos de uso dos actores ascendentes.
 - ▣ **Em qualquer local onde o ascendente seja esperado, podemos substituí-lo pelo descendente. (princípio da substituição)**

Nota: Se o princípio da substituição falhar, o mais certo é estar a generalizar o que não deve ser generalizado!

Generalização entre actores: quando não usar

64

- O objectivo da generalização foi simplificar o modelo
- Se o modelo, com a generalização, fica mais complicado, não deve generalizar

Lembre-se sempre: os modelos servem, entre outras coisas, para **comunicar** de forma eficaz com alguém.

-Use a generalização se o modelo resultante ficar mais simples de compreender.

-Não a use se isso complicar o modelo.

Relações entre casos de uso

Generalização

Inclusão

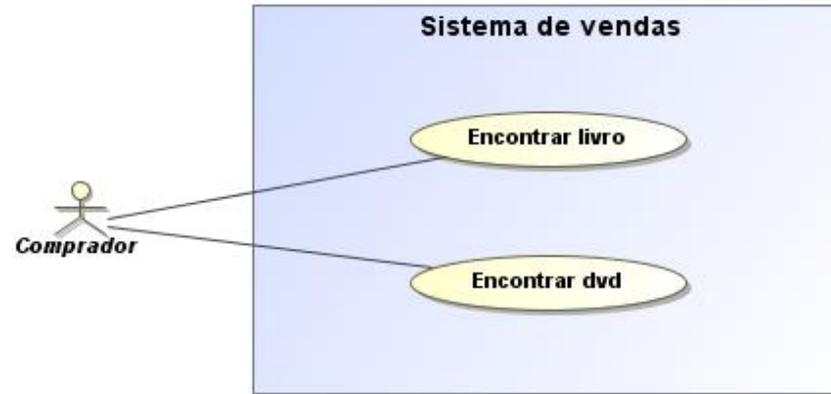
Extensão

Generalização de casos de uso

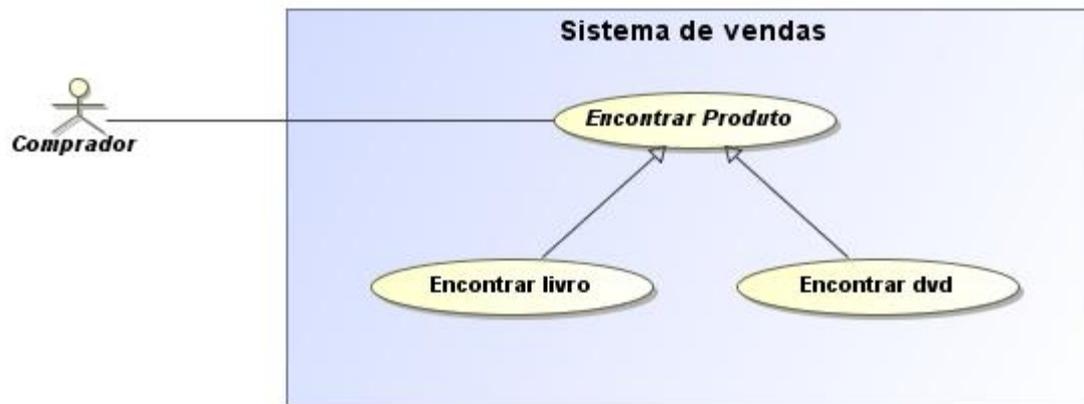
Generalização de casos de uso

67

- Em vez de:



- Podemos ter:



Generalização de casos de uso

68

- Quando temos casos de uso que são versões mais especializações de outros mais genéricos, podemos usar a generalização
- O objectivo deve ser sempre manter o modelo tão simples quanto possível
- Na generalização de casos de uso, o descendente representa uma versão mais específica do antecessor
- O descendente pode:
 - ▣ Herdar as características do ascendente
 - ▣ Adicionar novas características
 - ▣ Alterar características herdadas

Especificação do caso de uso abstracto

Encontrar Produto

69

| |
|--|
| Caso de Uso: <i>Encontrar produto</i> |
| Descrição: O comprador procura um produto no sistema |
| Actor principal: <i>Comprador</i> |
| Actores secundários: Nenhum |
| Pre-condições: Nenhuma |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o comprador selecciona a opção encontrar produto.2. O sistema pede ao comprador para indicar o critério de pesquisa a usar.3. O comprador introduz os critérios de pesquisa.4. O sistema procura produtos que satisfazem os critérios de pesquisa do comprador.5. Se o sistema encontra produtos que satisfazem os critérios<ol style="list-style-type: none">5.1. O sistema apresenta uma lista de produtos que satisfazem os critérios.6. Caso contrário<ol style="list-style-type: none">6.1. O sistema indica ao comprador que nenhum produto foi encontrado. |
| Pos-condições: Nenhuma |
| Fluxos alternativos: Nenhum |

Especificação do caso de uso Encontrar Livro

70

Caso de Uso: Encontrar Livro

Descrição: O comprador procura um livro no sistema

Actor principal: *Comprador*

Actores secundários: Nenhum

Pre-condições: Nenhuma

Fluxo principal:

1. (o1.) O caso de uso começa quando o comprador selecciona a opção encontrar livro.
2. (o2.) O sistema pede ao comprador para indicar o critério de pesquisa do livro a usar, que deve incluir o autor, título, ISBN, ou tópico.
3. O comprador introduz os critérios de pesquisa.
4. (o4.) O sistema procura livros que satisfazem os critérios de pesquisa do comprador.
5. (o5.) Se o sistema encontra livros que satisfazem os critérios
 - 5.1. O sistema mostra o best seller.
 - 5.2. (o5.1.) O sistema apresenta uma lista com até 5 livros que satisfazem os critérios.
 - 5.3. Para cada livro o sistema mostra o título, autor, preço e ISBN
 - 5.4. Enquanto houver mais livros para mostrar, o sistema dá ao comprador a possibilidade de pedir a próxima página com mais livros.
6. (6.) Caso contrário
 - 6.1. O sistema mostra o best seller.
 - 6.2. (6.1.) O sistema indica ao comprador que nenhum produto foi encontrado.

Pos-condições: Nenhuma

Fluxos alternativos: Nenhum

Especificação do caso de uso Encontrar dvd

71

| |
|--|
| Caso de Uso: Encontrar dvd |
| Descrição: O comprador procura um dvd no sistema |
| Actor principal: <i>Comprador</i> |
| Actores secundários: Nenhum |
| Pre-condições: Nenhuma |
| Fluxo principal: <ol style="list-style-type: none">1. (o1.) O caso de uso começa quando o comprador selecciona a opção encontrar livro.2. (o2.) O sistema pede ao comprador para indicar o critério de pesquisa do livro a usar, que deve incluir o autor, título, ISBN, ou tópico.3. O comprador introduz os critérios de pesquisa.4. (o4.) O sistema procura livros que satisfazem os critérios de pesquisa do comprador.5. (o5.) Se o sistema encontra livros que satisfazem os critérios<ol style="list-style-type: none">5.1. O sistema mostra o best seller.5.2. (o5.1.) O sistema apresenta uma lista com até 5 livros que satisfazem os critérios.5.3. Para cada livro o sistema mostra o título, autor, preço e ISBN5.4. Enquanto houver mais livros para mostrar, o sistema dá ao comprador a possibilidade de pedir a próxima página com mais livros.6. (6.) Caso contrário<ol style="list-style-type: none">6.1. O sistema mostra o best seller.6.2. (6.1.) O sistema indica ao comprador que nenhum produto foi encontrado. |
| Pos-condições: Nenhuma |
| Fluxos alternativos: Nenhum |

Generalização de casos de uso

72

A generalização de casos de uso factoriza comportamento comum de um ou mais casos de uso num caso de uso ascendente.

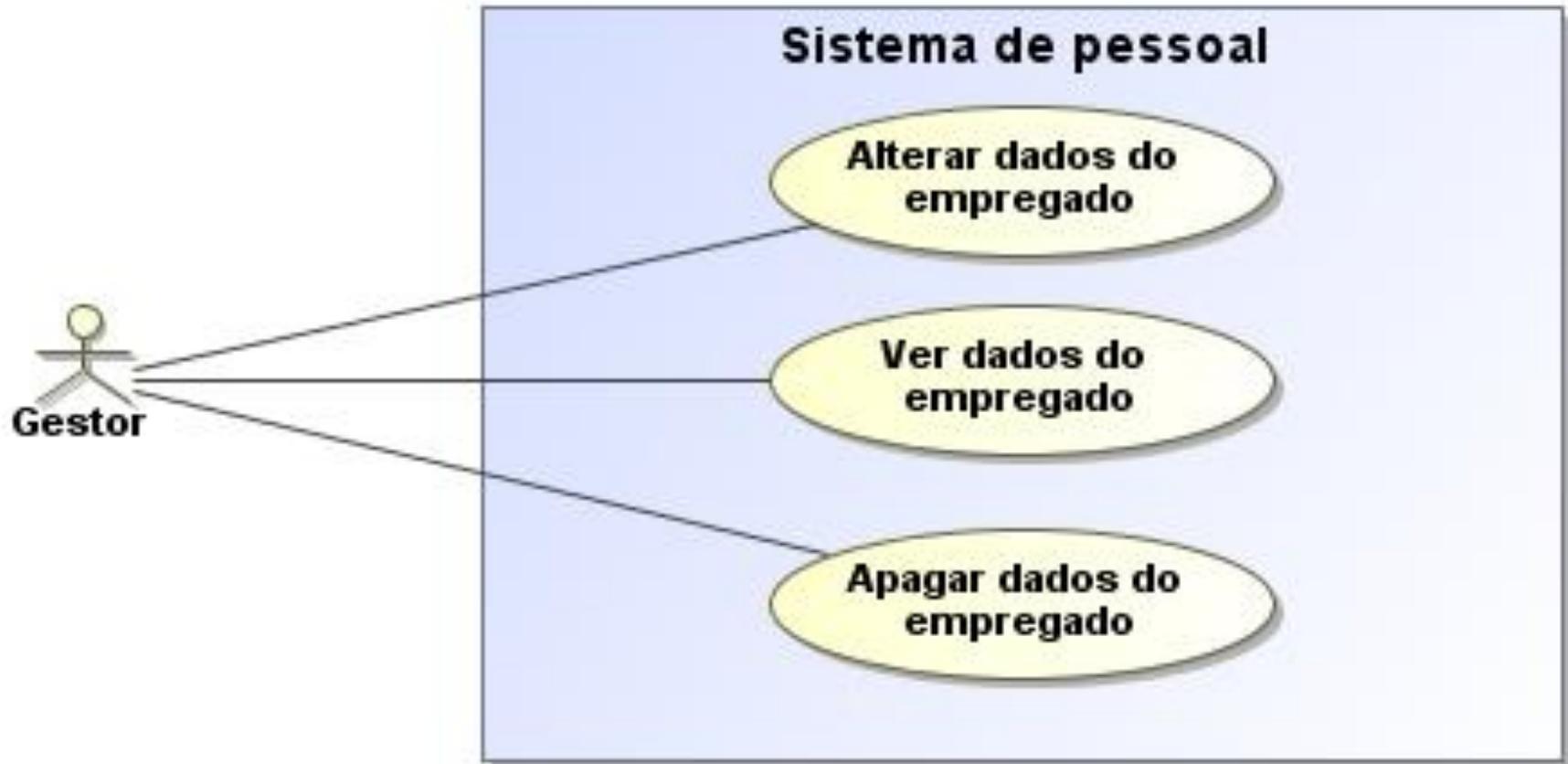
| Característica | Herdar | Adicionar | Redefinir |
|----------------------------|--------|-----------|-----------|
| Relação | Sim | Sim | Não |
| Ponto de extensão | Sim | Sim | Não |
| Pré-condição | Sim | Sim | Sim |
| Pós-condição | Sim | Sim | Sim |
| Passo no cenário principal | Sim | Sim | Sim |
| Cenário alternativo | Sim | Sim | Sim |

73

Reutilização de casos de uso

Considere um sistema de gestão de pessoal

74



Alterar dados do empregado

75

| |
|--|
| Caso de Uso: Alterar dados do empregado |
| Descrição: O gestor altera os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o gestor selecciona a opção alterar dados do empregado.2. O gestor introduz a identificação do empregado.3. O sistema encontra os detalhes do empregado.4. O sistema mostra os detalhes do empregado5. O gestor altera os dados do empregado <p>...</p> |
| Pós-condições: Os dados do empregado foram actualizados |
| Fluxos alternativos: Nenhum |

Ver dados do empregado

76

| |
|---|
| Caso de Uso: Ver dados do empregado |
| Descrição: O gestor vê os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o gestor selecciona a opção ver dados do empregado.2. O gestor introduz a identificação do empregado.3. O sistema encontra os detalhes do empregado.4. O sistema mostra os detalhes do empregado ... |
| Pós-condições: Nenhuma |
| Fluxos alternativos: Nenhum |

Apagar dados do empregado

77

| |
|--|
| Caso de Uso: Apagar dados do empregado |
| Descrição: O gestor apaga os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o gestor selecciona a opção apagar dados do empregado.2. O gestor introduz a identificação do empregado.3. O sistema encontra os detalhes do empregado.4. O sistema mostra os detalhes do empregado5. O gestor apaga os dados do empregado ... |
| Pós-condições: Os dados do empregado foram apagados. |
| Fluxos alternativos: Nenhum |

Identificação de um caso de uso reutilizável

78

| |
|--|
| Caso de Uso: Alterar dados do empregado |
| Descrição: O gestor altera os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: 1. O caso de uso começa quando o gestor selecciona a opção alterar dados do empregado. 2. O gestor introduz a identificação do empregado. 3. O sistema encontra os detalhes do empregado. 4. O sistema mostra os detalhes do empregado 5. O gestor altera os dados do empregado ... |
| Pós-condições: Os dados do empregado foram actualizados |
| Fluxos alternativos: Nenhum |

Nos três casos encontramos uma subsequência comum (na verdade, ela até é mais longa, mas neste caso estamos interessados em reutilizar apenas a pesquisa de empregado). Podemos factorizar o nosso modelo, para evitar repetições desnecessárias.

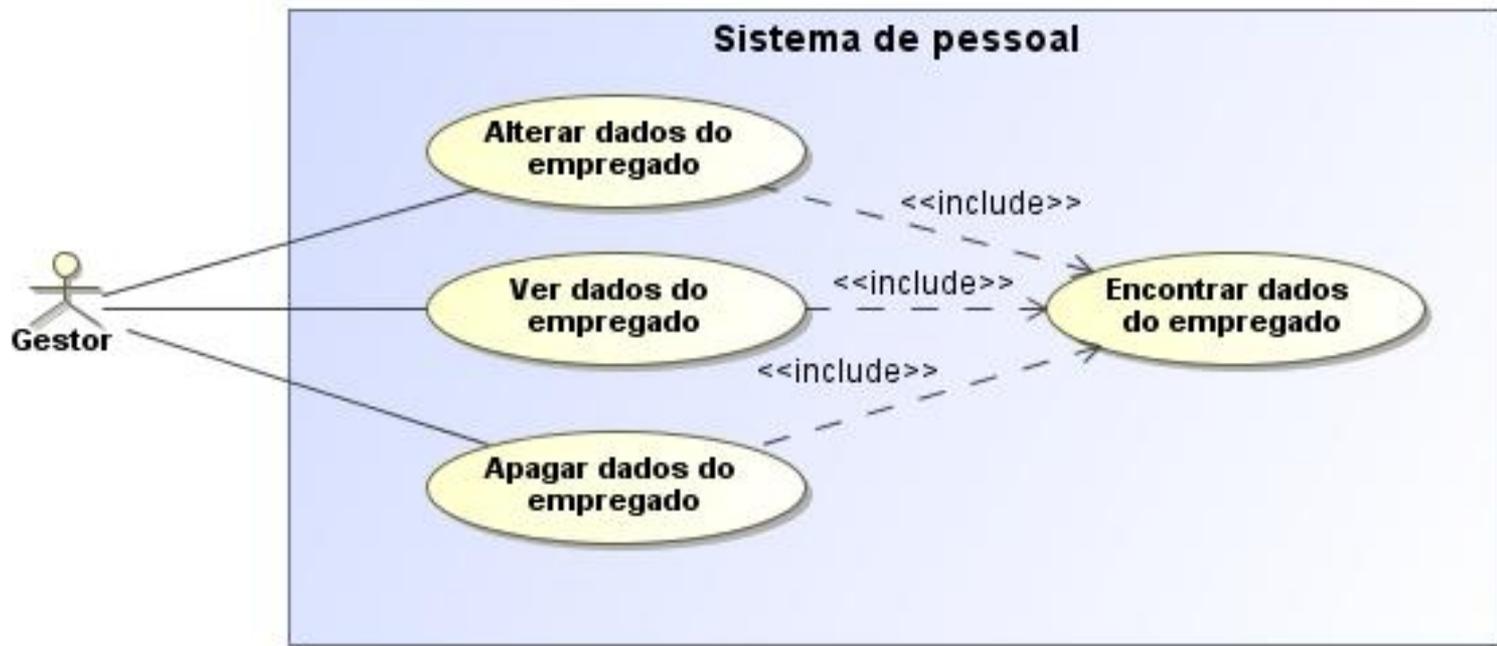
| |
|--|
| Caso de Uso: Ver dados do empregado |
| Descrição: O gestor vê os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: 1. O caso de uso começa quando o gestor selecciona a opção ver dados do empregado. 2. O gestor introduz a identificação do empregado. 3. O sistema encontra os detalhes do empregado. 4. O sistema mostra os detalhes do empregado ... |
| Pós-condições: Nenhuma |
| Fluxos alternativos: Nenhum |

| |
|--|
| Caso de Uso: Apagar dados do empregado |
| Descrição: O gestor apaga os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: 1. O caso de uso começa quando o gestor selecciona a opção apagar dados do empregado. 2. O gestor introduz a identificação do empregado. 3. O sistema encontra os detalhes do empregado. 4. O sistema mostra os detalhes do empregado 5. O gestor apaga os dados do empregado ... |
| Pós-condições: Os dados do empregado foram apagados. |
| Fluxos alternativos: Nenhum |

Reutilização de casos de uso

79

- A relação <<include>> permite factorizar o caso de uso de encontrar os dados do empregado, que depois é reutilizado pelos restantes casos de uso.



Alterar dados do empregado (nova versão)

80

| |
|---|
| Caso de Uso: Alterar dados do empregado |
| Descrição: O gestor altera os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o gestor selecciona a opção alterar dados do empregado.2. Include (Encontrar detalhes do empregado)3. O sistema mostra os detalhes do empregado4. O gestor altera os dados do empregado... |
| Pós-condições: Os dados do empregado foram actualizados |
| Fluxos alternativos: Nenhum |

Ver dados do empregado (nova versão)

81

| |
|--|
| Caso de Uso: Ver dados do empregado |
| Descrição: O gestor vê os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o gestor selecciona a opção ver dados do empregado.2. Include (Encontrar detalhes do empregado)3. O sistema mostra os detalhes do empregado ... |
| Pós-condições: Nenhuma |
| Fluxos alternativos: Nenhum |

Apagar dados do empregado (nova versão)

82

| |
|---|
| Caso de Uso: Apagar dados do empregado |
| Descrição: O gestor apaga os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o gestor selecciona a opção apagar dados do empregado.2. Include (Encontrar detalhes do empregado)3. O sistema mostra os detalhes do empregado4. O gestor apaga os dados do empregado <p>...</p> |
| Pós-condições: Os dados do empregado foram apagados. |
| Fluxos alternativos: Nenhum |

Caso de uso Encontrar detalhes do empregado

83

| |
|--|
| Caso de Uso: Encontrar dados do empregado |
| Descrição: O gestor encontra os dados referentes a um empregado |
| Actor principal: Gestor |
| Actores secundários: Nenhum |
| Pre-condições: O gestor está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O gestor introduz a identificação do empregado.2. O sistema encontra os detalhes do empregado. |
| Pós-condições: Nenhuma. |
| Fluxos alternativos: Nenhum |

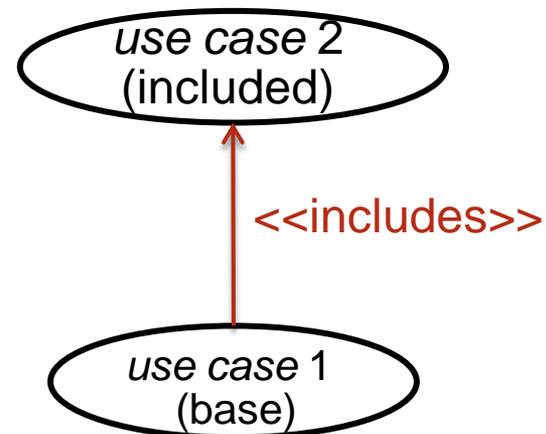
Associação <<includes>>

84

- Para promover a reutilização, partes idênticas de casos de uso(base) podem ser extraídas para criar um novo caso de uso
- Casos de uso (base) são depois relacionados com o novo caso de uso usando a associação <<includes>>
- O novo caso de uso pode não ter significado por si só

Vantagens:

- Rapidez em especificar
- Eliminação de comportamento “espalhado”
- Mais modularização
- Mais localização



Associação <<includes>>

85

- O caso de uso incluído fornece comportamento ao caso de uso base
- Modelo de execução:
 - ▣ O caso de uso base executa-se até ao ponto de inclusão;
 - ▣ Seguidamente, executa-se o caso de uso incluído;
 - ▣ Finalmente, a execução volta ao caso de uso base, para continuar no passo seguinte
- O caso de uso base apenas fica completo com o caso de uso incluído
- Se o caso de uso incluído é incompleto – chamamos-lhe um fragmento de comportamento
 - ▣ O caso de uso incluído não é iniciado directamente por actores
- No entanto, podemos também incluir casos de uso completos
 - ▣ Nesse caso, o caso de uso pode ser iniciado directamente por actores

Extensão de casos de uso

Extensão de casos de uso

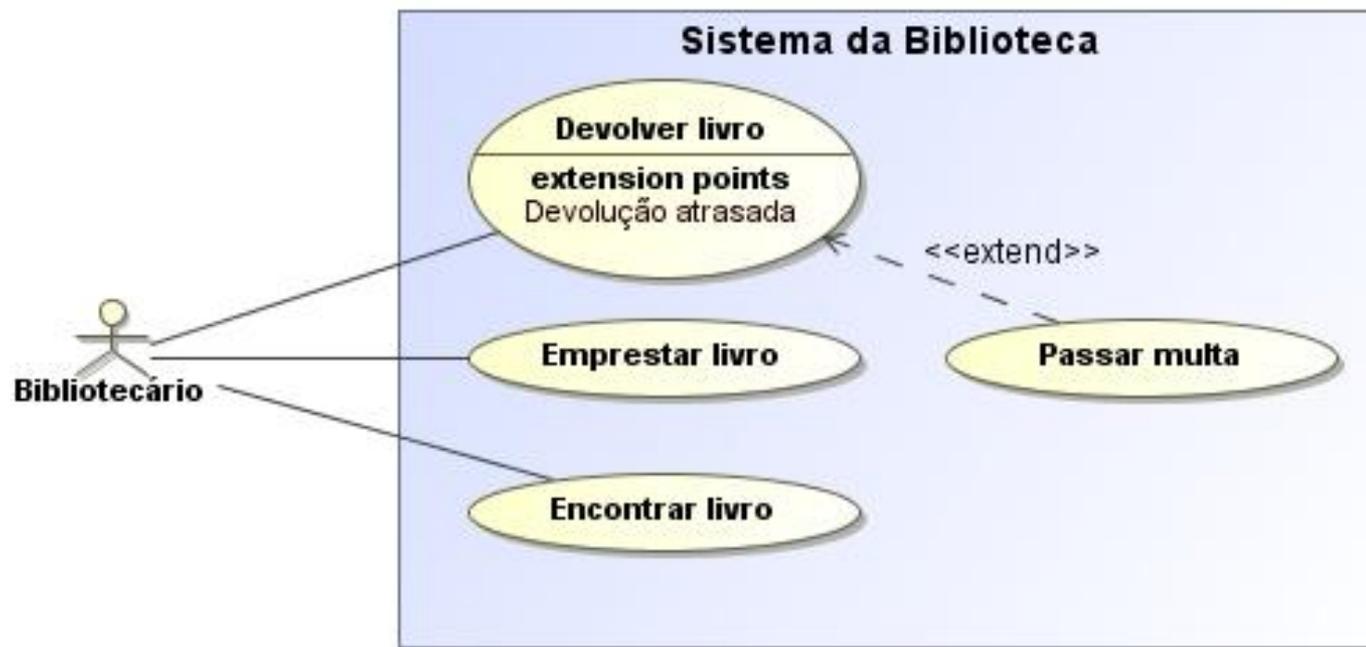
87

- Por vezes, queremos acrescentar novo comportamento a um caso de uso
 - Podemos definir nele pontos de extensão
 - Definimos também um caso de uso de extensão, que especifica os fragmentos de comportamento a inserir nos pontos de extensão
 - O caso de base não precisa de saber nada sobre os casos de uso que o estendem
 - Note que isto é completamente diferente de um include, em que o caso base fica incompleto sem o caso incluído

Considere um sistema de gestão de uma biblioteca

88

- **Passar multa** acrescenta comportamento a **devolver livro**, mas **devolver livro** pode perfeitamente ser executado sem que seja **passada a multa**



Especificação do caso de uso base

89

| |
|--|
| Caso de Uso: Devolver livro |
| Descrição: O bibliotecário devolve um livro emprestado |
| Actor principal: Bibliotecário |
| Actores secundários: Nenhum |
| Pre-condições: O bibliotecário está autenticado no sistema |
| Fluxo principal: <ol style="list-style-type: none">1. O caso de uso começa quando o bibliotecário selecciona a opção devolver livro.2. O bibliotecário introduz o identificador da pessoa que pediu o livro emprestado no sistema.3. O sistema apresenta os dados da pessoa que tinha o livro emprestado, incluindo informação sobre os livros que lhe foram emprestados4. O bibliotecário localiza o livro a devolver, na lista de livros emprestados <p>Extension point: devolução atrasada</p> <ol style="list-style-type: none">5. O bibliotecário marca o livro como devolvido <p>...</p> |
| Pós-condições: O livro foi devolvido. |
| Fluxos alternativos: Nenhum |

Especificação do caso de uso de extensão

90

| |
|---|
| Caso de Uso: Passar multa |
| Descrição: O bibliotecário regista e imprime uma multa |
| Actor principal: Bibliotecário |
| Actores secundários: Nenhum |
| Pre-condições do segmento 1: A devolução do livro está atrasada |
| Fluxo no segmento 1: <ol style="list-style-type: none">1. O bibliotecário insere no sistema os detalhes da multa (identificador do leitor e quantia a pagar)2. O sistema imprime a multa. |
| Pós-condições do segmento 1: A multa foi registada no sistema. O sistema imprimiu a multa |

O mecanismo de extensão

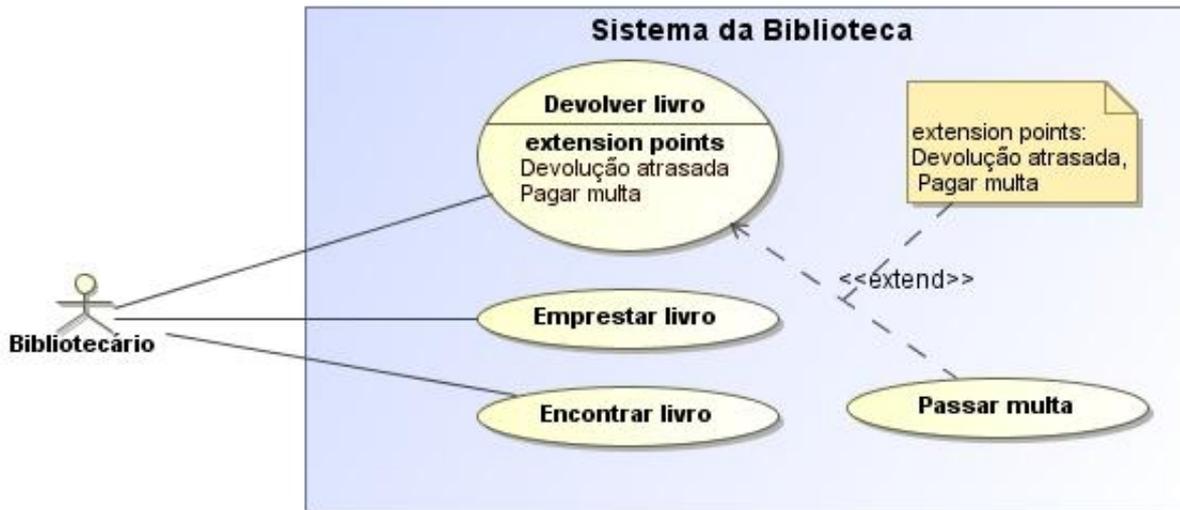
91

- A relação <<extend>> tem de especificar um ou mais pontos de extensão no caso de uso base
 - ▣ Se não o fizer, assume-se que se refere a todos os pontos de extensão
- O caso de uso de extensão tem de ter tantos segmentos quantos os pontos de extensão especificados na relação <<extend>>
- Podem existir vários casos de uso de extensão para um mesmo ponto de extensão, no mesmo caso base
 - ▣ Nesses casos, as várias extensões são executadas, mas por ordem indeterminada.
- Os casos de uso de extensão podem ter pré e pós condições
- Os casos de uso de extensão podem ser, também eles, estendidos
 - ▣ Tipicamente, **não queremos estender extensões**, porque o modelo tende a ficar complicado demais

Usando múltiplas extensões

92

- Suponha agora que o sistema deve suportar o pagamento de multas por atrasos na devolução de livros



Mecanismo útil para quando necessitamos de estender o caso base em vários pontos distintos. Neste exemplo, ao detectar que a devolução está atrasada, poderíamos modelar, no caso base, que verificaríamos todos os atrasos desse utente, para que ele apenas pagasse uma multa com o valor total, em vez de várias com valores parciais.

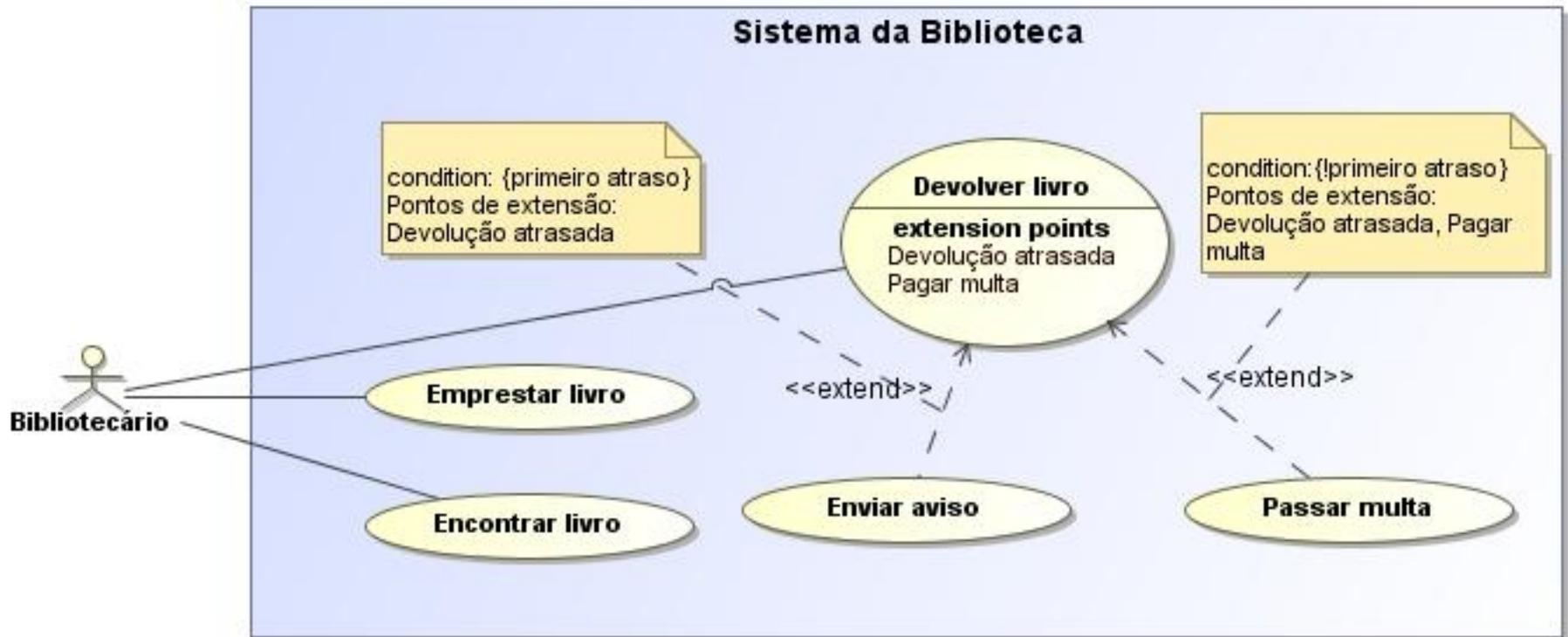
Especificação do caso de uso de extensão

93

| |
|---|
| Caso de Uso: Passar multa |
| Descrição: Segmento 1: O bibliotecário regista e imprime uma multa Segmento 2: O bibliotecário aceita o pagamento de uma multa |
| Actor principal: Bibliotecário |
| Actores secundários: Nenhum |
| Pré-condições do segmento 1: A devolução do livro está atrasada |
| Fluxo no segmento 1: <ol style="list-style-type: none">1. O bibliotecário insere no sistema os detalhes da multa (identificador do leitor e quantia a pagar)2. O sistema imprime a multa. |
| Pós-condições do segmento 1: A multa foi registada no sistema. O sistema imprimiu a multa. |
| Pré-condições no segmento 2: Existe uma multa por pagar pelo utente. |
| Fluxo no segmento 2: <ol style="list-style-type: none">1. O bibliotecário aceita o pagamento da multa.2. O bibliotecário introduz no sistema os dados da multa paga (identificador do leitor e quantia paga).3. O sistema imprime um recibo pelo pagamento da multa. |
| Pós-condições do segmento 2: <ol style="list-style-type: none">1. A multa fica registada como paga.2. O sistema imprimiu um recibo pela multa. |

Extensões condicionais

94



Especificação do caso de uso enviar aviso

95

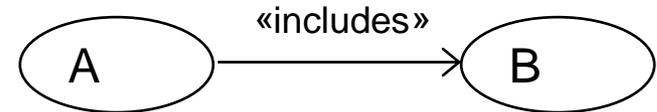
| |
|---|
| Caso de Uso: Enviar aviso |
| Descrição: Segmento 1: O bibliotecário envia um aviso de atraso. |
| Actor principal: Bibliotecário |
| Actores secundários: Nenhum |
| Pré-condições do segmento 1: A devolução do livro está atrasada |
| Fluxo no segmento 1: 1. O bibliotecário insere no sistema os detalhes da multa (identificador do leitor e quantia a pagar) |
| Pós-condições do segmento 1: O aviso fica registado no sistema. |

Relações entre casos de uso (sumário)

96

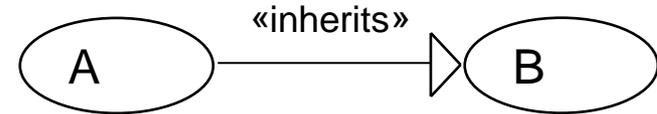
□ “includes”

- o comportamento de **B** é incluído em **A**
- o caso de uso **B** incluído é necessário para assegurar a funcionalidade do **A**
- **A** conhece a existência de **B**
- **A** é o caso de uso base



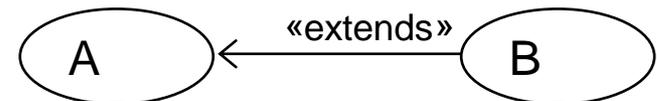
□ “inherits”

- idêntico à herança entre classes
- **A** herda o comportamento de **B** e pode redefini-lo ou estendê-lo
- é possível definir use cases abstractos
- **B** é o super- caso de uso



□ “extends”

- o comportamento de **B** pode vir a estender o comportamento de **A**
- **A** não conhece a existência de **B**
- **A** é o caso de uso base



97

Estilo de utilização

Sobre a utilização de mecanismos avançados na definição de casos de uso

98

- Tipicamente, os interlocutores conseguem entender as noções de actor e de caso de uso com relativamente pouco treino
- Têm mais dificuldade em compreender a noção de generalização de actores
- A utilização excessiva de relações <<include>> torna os modelos difíceis de compreender, porque é necessário analisar vários casos de uso
- O mecanismo de extensão é normalmente difícil de compreender por interlocutores sem formação específica em modelação (e gera dificuldades mesmo entre muitos engenheiros informáticos ☹)
- A generalização de casos de uso deve ser feita usando casos de uso abstractos – a generalização de casos de uso concretos como base torna os casos de uso descendentes (filhos) difíceis de compreender

Estilo da descrição textual

99

- Mantenha a descrição dos casos de uso pequena (menos de uma página)
- Coloque o ênfase no quê e não em como
 - ▣ Deixe os detalhes de desenho... para o desenho!
- Evite a decomposição funcional de casos de uso
 - ▣ Um erro frequente é começar por criar um caso de uso demasiado genérico e depois ir decompondo, até se chegar a casos de uso mais “primitivos” que são, finalmente, de uma granularidade adequada para que possam ser detalhados

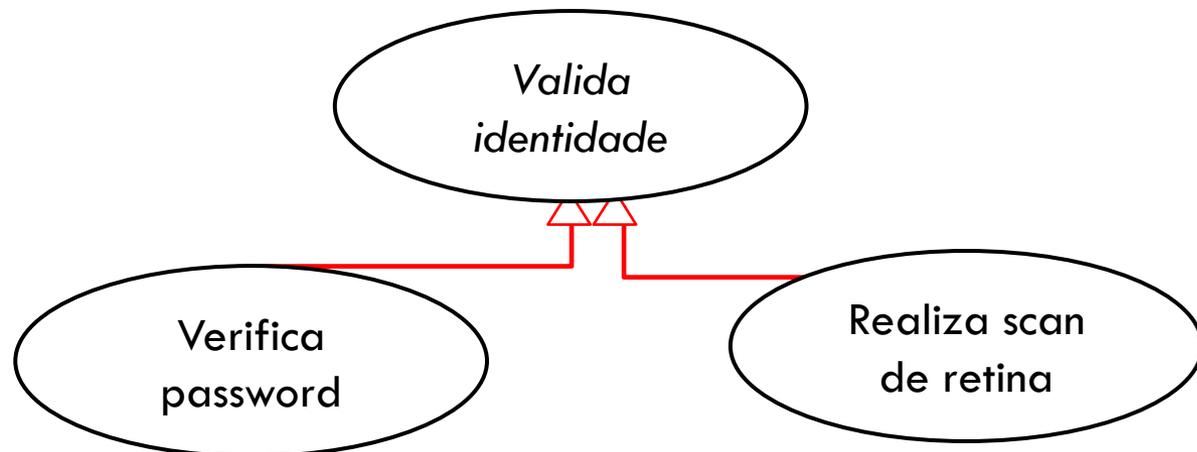
100

Mais alguns exemplos...

<<inherits>> entre casos de uso

101

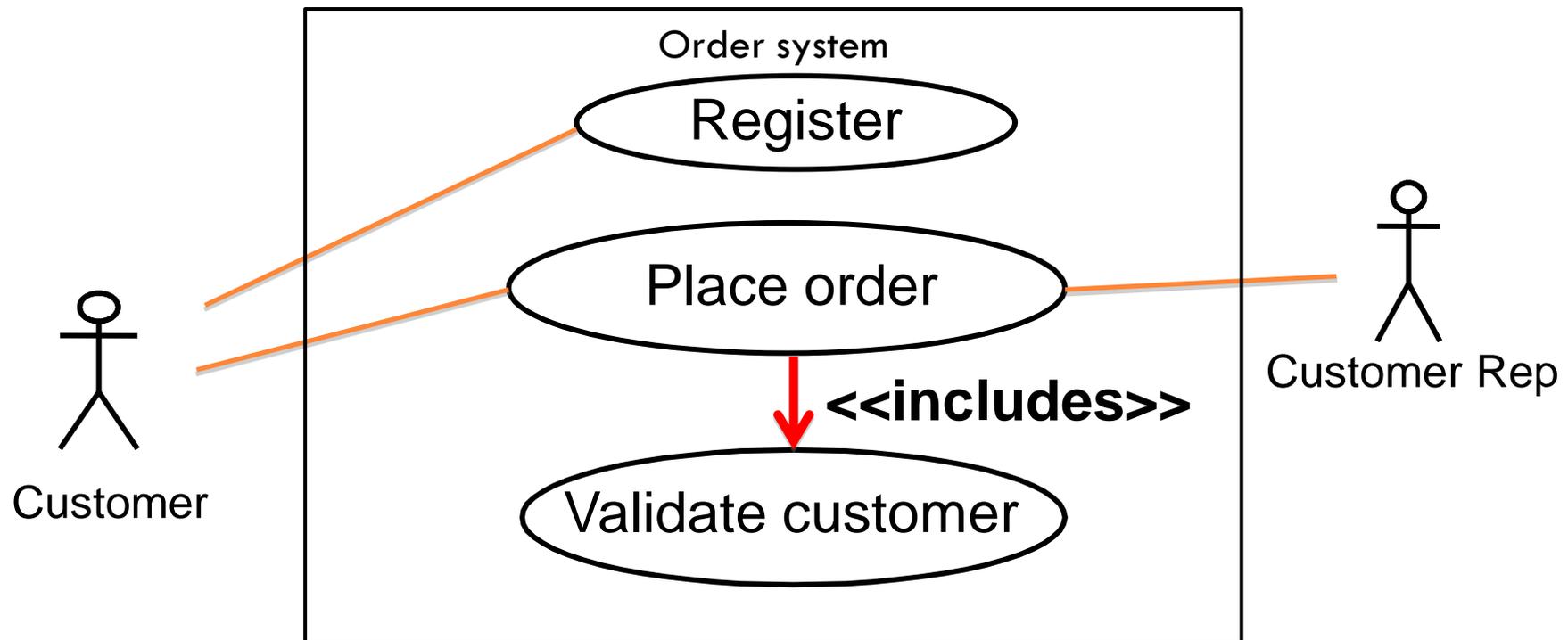
- O que fazer se o actor:
 - ▣ Puder usar qualquer das formas alternativas de validação?
 - ▣ Tiver de usar uma delas em particular?
 - ▣ Tiver de usar as duas?



<<includes>>: exemplo

102

- E se os clientes tiverem que se registar?
- Qual a consequência para Place Order?

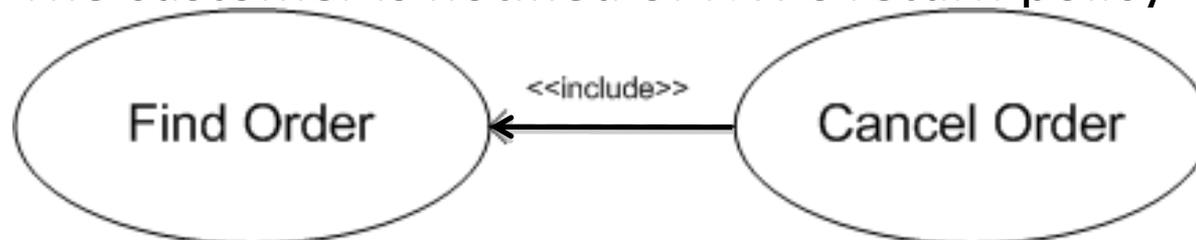


Cancel Order Use Case

103

1. The use case begins when a customer requests “cancel order”
2. <<include>> Find Order
3. If the order status is Confirmed
 - a) The order is marked Cancelled
 - b) The accounting system is notified to credit the customer’s account and the use case ends.
4. If the order status is Shipped
 - a) The customer is notified of NW’s return policy

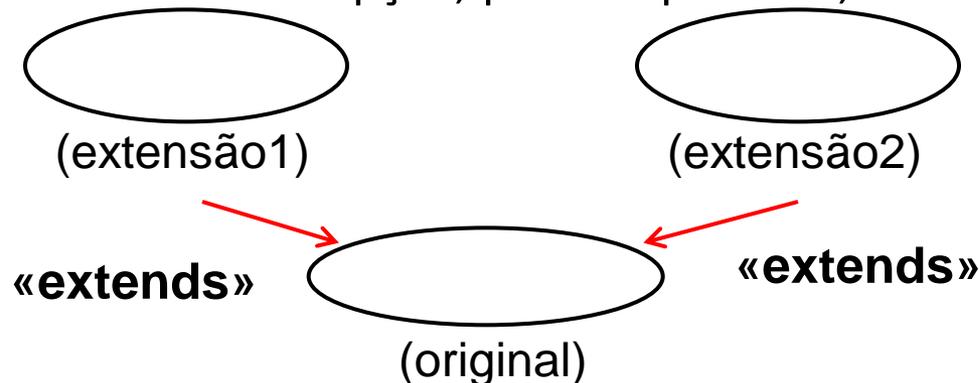
**Modularização
Reutilização
(rapidez)**



Associação “extends”

104

- A associação <<extends>> relaciona como um caso de uso pode ser estender o comportamento de outro.
- O caso de uso original não precisa de saber sobre a existência das extensões. É construído independentemente.
- Os casos de uso que estendem outros podem ser vistos como interrupções do original
 - ▣ para definir casos de exceção, partes opcionais, alternativas



Alguns cenários secundários podem representar-se como extensões: exemplo

105

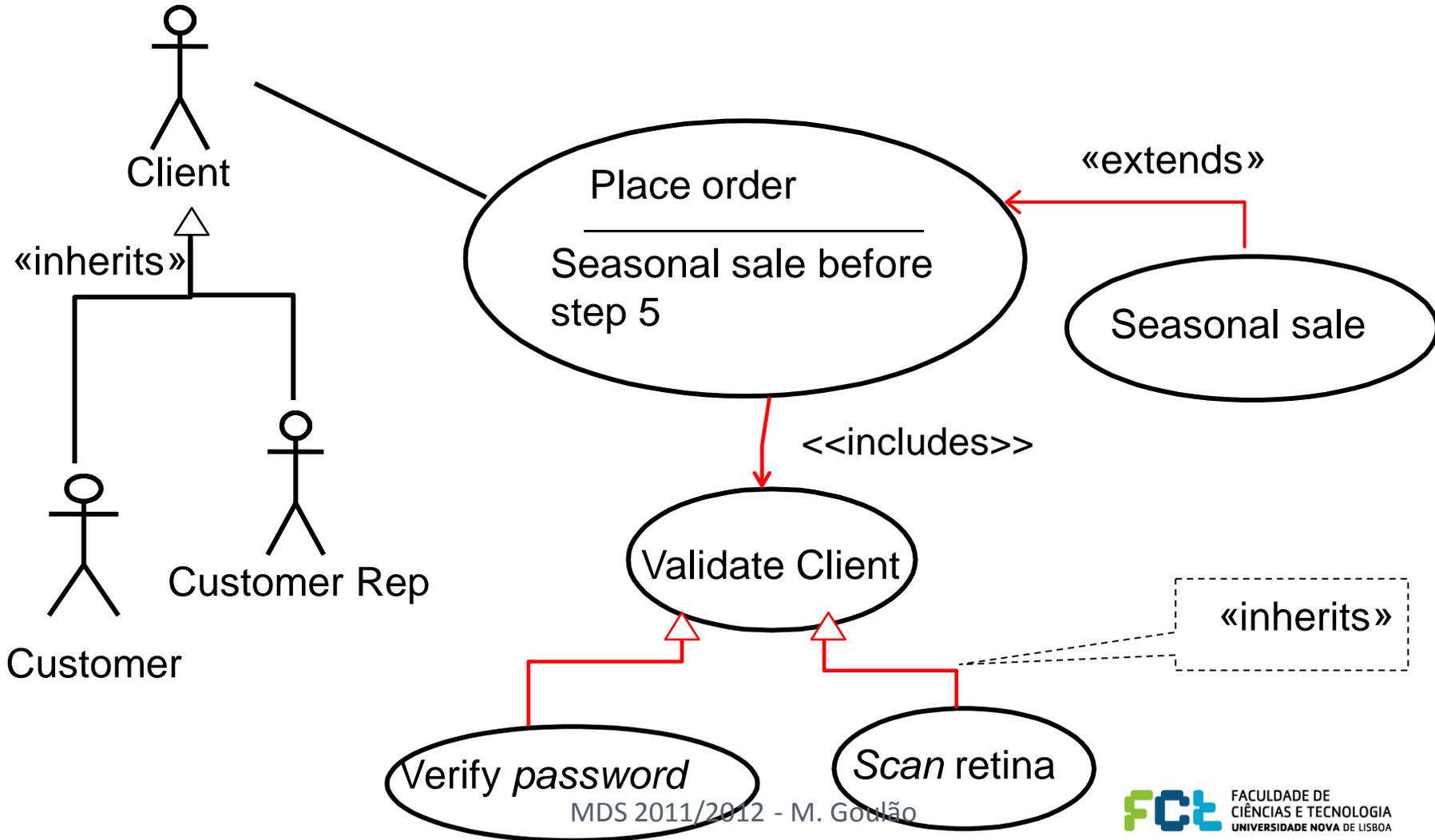
□ Fluxo de eventos do cenário secundário Seasonal Sale para o caso de uso Place Order

1. The system will get the sale discount for the product
2. The system will display the discount on the order
3. The system will calculate a discount amount
4. The system will subtract the discount amount from the order total

- Este cenário está relacionado ao principal através de *extension points*
 - ◆ “Place order” *extension points*: “Seasonal sale”, antes do passo 5 (ver diagrama)
 - ◆ São representados nos diagramas de *use cases*

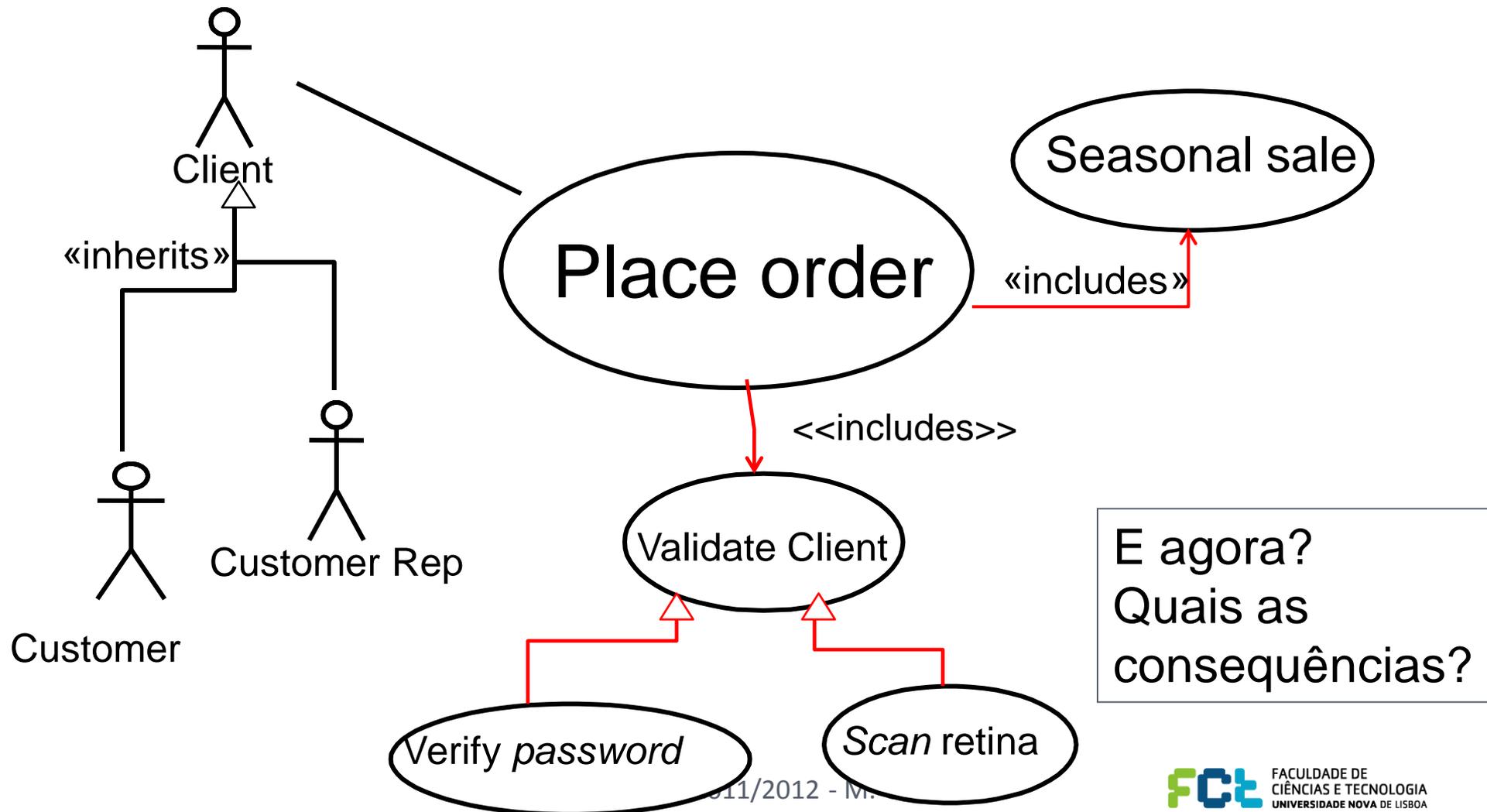
Modelo UCs com os três tipos de relação

106



Mas, e se Seasonal Sale se relacionasse com <<includes>>?

107



Place Order Use Case

108

Pre-condition:

A valid user has logged into the system

Flow of Events:

Basic Path:

1. The use case starts when the customer selects place order.
2. The customer enters his/her name and address
3. **<<includes>> Validate Client**
4. If the customer enters only the zip code the system will supply the city and the state.
5. **<<includes>> Seasonal sale**
6. The customer will enter product codes for desired products.
7. The system will supply a product description and the price for each item.

8. The system will keep running total of items ordered as they are entered.
9. The customer will enter credit card payment information.
10. The customer will select submit.
11. The system will verify the information, save the order as pending, and forward the payment information to the accounting system.
12. When payment is confirmed, the order is marked Confirmed, an order ID is returned to the customer, and the use case end.

Post Condition:

The order has been saved in the system and marked Confirmed.



Má decisão!!!

Recapitulando:

Para construir um modelo de casos de uso

109

- Identificar os actores que interagem com o sistema
- Organizar (hierarquicamente) os actores
- Para cada actor identificar use cases
- Para cada use case identificar o cenário principal e os secundários
- Descrever cada cenário
- Organizar os use cases utilizando as relações extends, includes e inherits
- Se existirem, acrescentar os actores receptores de informação